Treatment Learning

August 31, 2006

Treatment Learning

- Less is more
- Don’t tell me *what is*, tell me *what to do*

- Data Mining for Busy People
  - Wikipedia: http://en.wikipedia.org/wiki/Data_Mining_For_Very_Busy_People
  - Original article: http://menzies.us/pdf/03tar2.pdf

Classifiers vs. Treatment Learners

- Standard data miners produce classifiers to *categorize* new examples.
- Classifiers are used for *recognition*.

- A treatment learner produces rules to *change* the expected class distribution.
- A treatment learner is used for planning some minimal *action* to improve the odds that new examples will belong to a desired class.

Classifiers vs. Treatment Learners (2)

- Classifiers are about representational accuracy.
  - If the target is complex then the resulting tree will be complex.

- Treatment learners are all about minimality.
  - What is the *least* you need to do to *most* affect something?

Example

WEKA's J48 Classifier

```
LSTAT <= 14.98
|  RM <= 6.54
|  |  DIS <= 1.6102
|  |  |  DIS <= 1.358: high (4.0/1.0)
|  |  |  DIS > 1.358
|  |  |  |  LSTAT <= 12.67: low (2.0)
|  |  |  |  LSTAT > 12.67: medlow (2.0)
|  |  DIS > 1.6102
|  |  |  TAX <= 222
|  |  |  |  CRIM <= 0.06888: medhigh (3.0)
|  |  |  |  CRIM > 0.06888: medlow (4.0)
|  |  |  TAX > 222: medlow (199.0/9.0)
|  RM > 6.54
|  |  RM <= 7.42
|  |  |  DIS <= 1.8773: high (4.0/1.0)
|  |  |  DIS > 1.8773
|  |  |  |  PTRATIO <= 19.2
|  |  |  |  |  RM <= 7.007
41: medhigh (25.0/1.0)
1: medlow (2.0)
54
61
<= 7.87: medhigh (9.0)
> 7.87: medlow (3.0/1.0)
1: medlow (3.0)
: medlow (14.0/1.0)
|  |  |  |  |  RM > 7.007: medhigh (29.0)
|  |  |  |  PTRATIO > 19.2: medlow (11.0/1.0)
|  |  RM > 7.42
|  |  |  PTRATIO <= 17.9: high (25.0/1.0)
|  |  |  PTRATIO > 17.9
|  |  |  |  AGE <= 43.7: high (2.0)
|  |  |  |  AGE > 43.7: medhigh (3.0/1.0)
LSTAT > 14.98
|  CRIM <= 0.63796
|  |  INDUS <= 25.65
|  |  |  DIS <= 1.7984: low (5.0/1.0)
dlow (37.0/2.0)
(4.0)
2: low (8.0)
: medlow (6.0/1.0)
dlow (5.0)
|  |  |  |  CHAS = 0: low (80.0/8.0)
|  |  |  |  CHAS = 1
|  |  |  |  |  DIS <= 1.7455: low (2.0)
|  |  |  |  |  DIS > 1.7455: medlow (2.0)
```

TAR3

Baseline

```
   low:~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~  [   133 - 29%]
 medlow:~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~  [   131 - 29%]
medhigh:~~~~~~~~~~~~~~~~~~~~~~           [    97 - 21%]
   high:~~~~~~~~~~~~~~~~~~~~~            [    94 - 21%]
```

Treatment

Treatment:[PTRATIO=[12.6..16) RM=[6.7..9.78)]

Result

```
   low:                                  [     0 -  0%]
 medlow:                                 [     0 -  0%]
medhigh:                                 [     1 -  3%]
   high:~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~  [    38 - 97%]
```

Lift

- What makes one treatment better than another? *Lift*.

- Treatment learners assess their theories by comparing:
  - A weighted sum of the classes in the *baseline*
  - A weighted sum of the classes in the prediction

- The sums are normalized so that the baseline has a lift of 1.
- The lift of the *predicted distribution*, based on the treatment from the previous slide, has a lift of 2.34.

Best Support

- Ideally treatments have large lift.
  - This can be achieved by making the treatment more specific.
- However, the more specific the treatment the more of the data it filters out.
  - The less data the treatment is based on, the less evidence there is to support that treatment's high lift value.
- To avoid overly specific treatments, treatment learners use *best support*: the percentage of best class instances supporting the treatment.
  - Treatments are rejected if they do not satisfy *minimal best support*.

The Small Treatment Effect

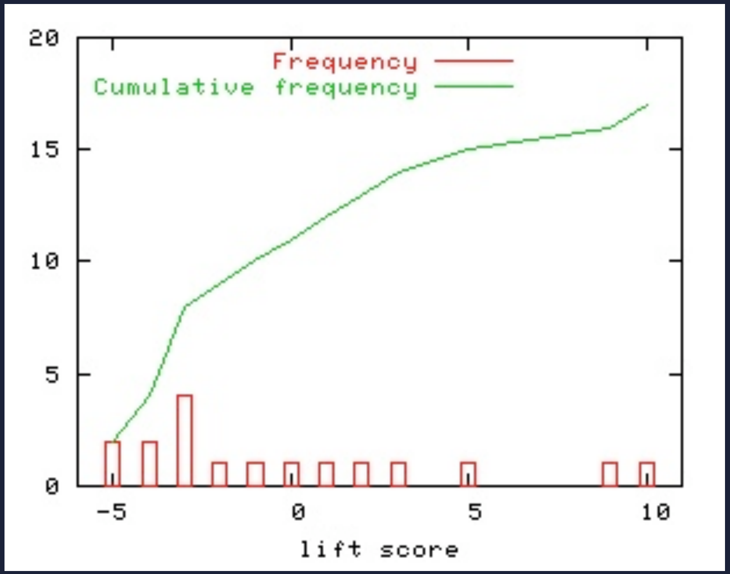
- A consequence of using the minimal best support criterion from the previous slide is that treatments are kept small.
  - Many best class instances support treatment → treatment isn't too specific → treatment doesn't have too many conjunctions → treatment is *small*.
- This is good!
  - Small treatments are easier to understand.
  - Small treatments are easier to implement.

Inside the TAR3 Treatment Learner

- *Classes* have weights (1, 2, 4, 8…).
- *Baseline best* = number of highest weighted instances in the unfiltered data.
- *Yield* = sum of (class frequency × weight) (i.e., lift before normalizing).
- *Baseline* = yield of all data.
- *Treatment* (*Rx*) = a conjunction of constraints on attributes.
- *Selected* = subset of data consistent with treatment.
- *Lift* = (yield of selected) / (yield of baseline).
  - Lift > 1 = better; lift < 1 = worse.
- Treatment (Rx) learning: seek smallest treatment
  - With highest lift (the controller – what to do)
  - With lowest lift (the monitor – what to avoid)
  - With enough support (e.g., 20% × baseline best)

Inside the TAR3 Treatment Learner (2)

- Build a treatment:
  - Randomly select *N* between 1 and user-specified *maxSize*.
  - Calculate the lift of each individual attribute value.
  - Search through combinations of highly scoring values.
  - Convert individual value scores into a cumulative probability distribution.
  - Build a treatment by selecting *N* values at random from this distribution.
  - Score treatment by summing class weights for all instances not filtered out by the treatment.
  - Ignore the treatment if its support is less than *bestClass*.



Inside the TAR3 Treatment Learner (3)

- Follow the procedure on the previous slide to produce *randomTrials* number of treatments.
- Delete all but the best *maxNumber* of them.

- Repeat this until, after a *futileTrials* number of attempts, no new treatments can be added to the top *maxNumber* of them.

Using TAR3

- Three input files are used. For example, the housing example above uses:
  - housing.data
    - CSV file, one instance per row; instance is a list of attribute values followed by a class.
  - housing.names
    - First line is classes in ascending order: *low, medlow, medhigh, high* (house cost).
    - Additional line for each attribute: *CRIM: continous* (CRIM is a numeric value representing crime rate). Discrete values can be used also: *mood: happy, sad*.
  - housing.cfg
    - Configuration information for the following values: *granularity*, *maxNumber*, *maxSize*, *randomTrials*, *futileTrials*, *bestClass*.

TAR3 Sample Data File

Class Value

Attribute Values

Instances

```
38.35180,0.00,18.100,0,0.6930,5.4530,100.00,1.4896,24,low
67.92080,0.00,18.100,0,0.6930,5.6830,100.00,1.4254,24,low
25.04610,0.00,18.100,0,0.6930,5.9870,100.00,1.5888,24,low
9.91655,0.00,18.100,0,0.6930,5.8520,77.80,1.5004,24,low
0.18337,0.00,27.740,0,0.6090,5.4140,98.30,1.7554,4,low
45.74610,0.00,18.100,0,0.6930,4.5190,100.00,1.6582,24,low
14.23620,0.00,18.100,0,0.6930,6.3430,100.00,1.5741,24,low
16.81180,0.00,18.100,0,0.7000,5.2770,98.10,1.4261,24,low
18.08460,0.00,18.100,0,0.6790,6.4340,100.00,1.8347,24,low
22.59710,0.00,18.100,0,0.7000,5.0000,89.50,1.5184,24,low
10.83420,0.00,18.100,0,0.6790,6.7820,90.80,1.8195,24,low
0.20746,0.00,27.740,0,0.6090,5.0930,98.00,1.8226,4,low
15.86030,0.00,18.100,0,0.6790,5.8960,95.40,1.9096,24,low
```

TAR3 Names and Configuration Files

Comments

Names File (.names)

Ranked Classes

```
low,medlow,medhigh,high      | These are the classes
CRIM:continuous              | This is a continous attribute
ZN:continuous
INDUS:continuous
CHAS:continuous
NOX:continuous
RM:continuous
AGE:continuous
DIS:continuous
RAD:continuous
TAX:continuous
```

Attribute Names, Types

Configuration File (.cfg)

```
granularity:2
maxNumber:10
maxSize:4
randomTrials:50
futileTrials:5
bestClass:50%
```

Configuration Settings

TAR3 Usage, Links

```
$ ls
housing.cfg    housing.data   housing.names  tar3

$ ./tar3 housing
```

- Executable TAR3: ~timm/bin/wisp/tar3
- Path to housing data set: ~timm/wisp/trunk/tar3/data/HOUSING/housing
- Path to my examples: ~dowen/public_html/tl/ (or www.csee.wvu.edu/~dowen/tl/)
- More data to play with: www.ics.uci.edu/~mlearn/MLRepository.html

Using TAR3 to Tune a Model

- As described in previous slides, you can use TAR3 to get concise information from a data set.
- You can also use TAR3 to tune a model.
  - Use the model to generate a data set.
    - Attributes are tunable parameters in the model.
    - Classes categorize some important output measure.
    - Generate a bunch of instances (systematic, random, whatever) with different parameter settings.
  - Run TAR3 on the data set.
  - Treatments predict optimal parameter settings.
- For example, see *~dowen/public_html/tl/cfgsim*…

Review Questions

- J48 produces *decision trees*. Describe decision trees.
- TAR3 produces *treatments*. Describe treatments and distinguish them from decision trees.
- List one application for decision tree learning and another for treatments. Carefully justify your answer.
- Describe the *small treatment effect* and explain why it occurs. Hint: lift, minimum best support, dumb apes get by…
- Explain each of the parameters of the TAR3 .cfg file. What would happen if each parameter was doubled? Halved? Hint: your answer might include some brief notes on TAR3's search for treatments.